Behavior Based Control For An Outdoor Crisis Management Robot

Daniela Doroftei, Geert De Cubber, Eric Colon, Yvan Baudoin

Royal Military Academy, Department of Mechanical Engineering (MSTA)
Av. de la Renaissance 30, 1000 Brussels
{daniela.doroftei, geert.de.cubber, eric.colon, yvan.baudoin}@rma.ac.be

Keywords: Stereo Vision, Terrain Traversability, Outdoor Mobile Robot Navigation

1. Introduction

Crisis management teams (e.g. fire & rescue services, anti-terrorist units, ..) are often confronted with dramatic situations where critical decisions have to be made within hard time constraints. In these circumstances, a complete overview of the crisis site is necessary to take correct decisions. However, obtaining such a complete overview of a complex site is not possible in real-life situations when the crisis management teams are confronted with large and complex unknown incident sites. In these situations, the crisis management teams typically concentrate their effort on a primary incident location (e.g. a building on fire, a wreckage, ...) and only after some time (depending on the manpower and the severity of the incident), they turn their attention towards the larger surroundings, e.g. searching for victims scattered around the incident site. A mobile robotic agent could aid in these circumstances, gaining valuable time by monitoring the area around the primary incident site while the crisis management teams perform their work. However, as the human crisis management teams are in general already overloaded with work and information in any medium or large scale crisis situation, it is essential that such a robotic agent - to be useful - does not require extensive human control (hence it should be semi-autonomous) and it should only report critical information back to the crisis management control center.

In the framework of the View-Finder project, such an outdoor mobile robotic platform is being developed. This semi-autonomous agent, shown on figure 1, was equipped with a differential GPS system for accurate geo-registered positioning, and a stereo vision system. In this paper, we discuss the development of the control architecture for this semi-autonomous outdoor mobile robot.

The behavior based control paradigm was chosen as a control mechanism due to its flexible nature, allowing the design of complex robot behavior through the integration of multiple relatively simple sensor-actuator relations. Through this control architecture, the robot must be able to search for human victims on an incident site, while navigating semi-autonomously, using stereo vision as a main source of sensor information. The design and development of a control architecture for such a robotic agent raises 3 main questions:

Keywords: Behavior-Based Navigation, Robot Control, Outdoor Mobile Robot Navigation
Funding: View-Finder FP6 IST 045541

1. How can we design the individual behaviors, such that the robot is capable of avoiding obstacles and of navigating semi-autonomously?
2. How can these individual behaviors be combined in an optimal, leading to a rational and coherent global robot behavior?
3. How can all these capabilities be combined in a comprehensive and modular framework, such that the robot can handle a high-level task (searching for human victims) with minimal input from human operators, by navigating in a complex, dynamic and environment, while avoiding potentially hazardous obstacles?

In this paper, we present each of these three main aspects of the general robot control architecture more in detail.

2. The Global Behavior-Based Robot Navigation Architecture

Figure 1 illustrates the general robot control architecture, set up as a test bed for the algorithms discussed in this paper. The RobuDem robot used in this setup features 2 on-board processing stations, one for low-level motor control (Syndex Robot Controller), and another one for all the high-level functions. A remote robot control PC is used to control the robot and to visualize the robot measurements (color images, victim data) from a safe distance. All data transfer between modules occurs via TCP and UDP-based connections, relying on the CORBA [8] and CoRoBa [1] protocols. The wireless link from the on-board high-level PC to the remote robot control PC is set up via a Wi-Max connection. To assure the quality of service over this wireless link, the use of the MailMan protocol is investigated.

A behavior-based navigational architecture is used for semi-autonomous intelligent robot control. Behavior-based techniques have gained a widely popularity in the robotics community [5], due to the flexible and modular nature of behavior-based controllers, facilitating the design process. Following the behavior based formalism, a complex control task is subdivided into a number of more simple modules, called behaviors, which each describe one aspect of the sensing, reasoning and actuation robot control chain. Each behavior outputs an objective function, $o_1(x)$, …, $o_n(x)$, which are multi-dimensional normalized functions of the output parameters, where $x = x_1, \ldots, x_n \in \mathbb{R}^n$ is an *n*-dimensional decision variable vector. The degree of attainment of a particular alternative *x*, with respect to the k^{th} objective is given by $o_k(x)$.

Recall that the RobuDem robot is equipped with two main sensing abilities: a stereo vision system and a GPS system. The information from the stereo vision system is used threefold:

1. The color images are sent over the wireless link, such that the human operator receives at all time a visual cue of the environment. This is absolutely necessary when the robot is operating under tele-operation mode.
2. The (left) color image is sent to a victim detection module. This module incorporates a vision-based human victim detector, as presented in [3]. The victim detection module will report any detected human victims back to the human operator at the remote control station.

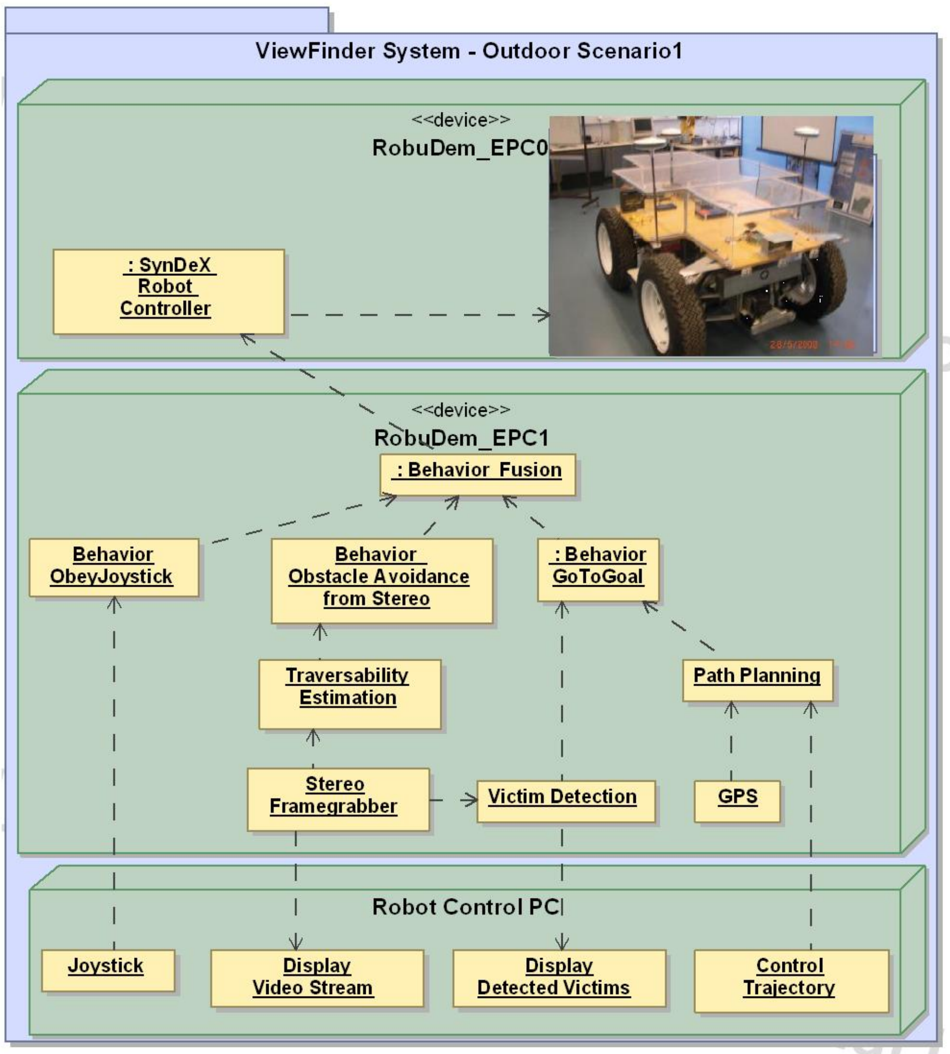

Figure 1: Robot Control Architecture

3. The calculated stereo disparity image is sent to a terrain traversability estimation module. This module incorporates a stereo-based terrain traversability analysis algorithm, as presented in [2]. From the obstacle map, a behavior is constructed to steer the robot away from obstacles.

The GPS system delivers accurate robot positioning information, which is sent to the operator at the remote control station. At the same time, this data is sent to a path planning module. From the robot control station, the human operator is able to compile a list of waypoints for the robot. The path planning module compares this list of waypoints with the robot position and calculates a trajectory to steer the robot to the goal positions in the list. The first point on this trajectory list is sent to a *GoToGoal* behavior module, which aims to steer the robot to this point, as such executing the trajectory defined by the path planner.

3. Behavior Design

In this section, we will investigate the design of the behaviors referenced on Figure 1. As mentioned before, a behavior is a function which relates sensor measurements to actions in the form of an objective function. In the case of robot control, the objective function of each behavior can be regarded as two-dimensional normalized function of robot steering velocity *v* and direction α. For this setup, three behaviors are defined which relate the abstract sensor information into robot actions. These three behaviors are:

1. *Obey Joystick Commands*. If desired, the human operator can control the robot by means of a joystick. The joystick commands are directly related to the robot steering angle and direction, so the transformation of the joystick control command into an objective function can be performed straightforward by calculating a two-dimensional Gaussian from the joystick input $v_{Joystick}, \alpha_{Joystick}$:

$$o_{Joystick}(v,\alpha) = \frac{1}{\sqrt{2\pi^2\sigma^4}} \exp\left(-\left(\frac{{v - v_{Joystick}}^2}{2\sigma^2} + \frac{{\alpha - \alpha_{Joystick}}^2}{2\sigma^2} \right) \right)$$

2. *Obstacle Avoidance Using Stereo*. To drive the robot away from obstacles detected by the terrain traversability analysis algorithm [], the obstacle image is analyzed. The depth values of pixels corresponding to obstacles are accumulated per vertical line in the image and the resulting function is inverted and normalized. This allows to deduce a function *f* of the viewing angle ⍰ as shown on figure 2.

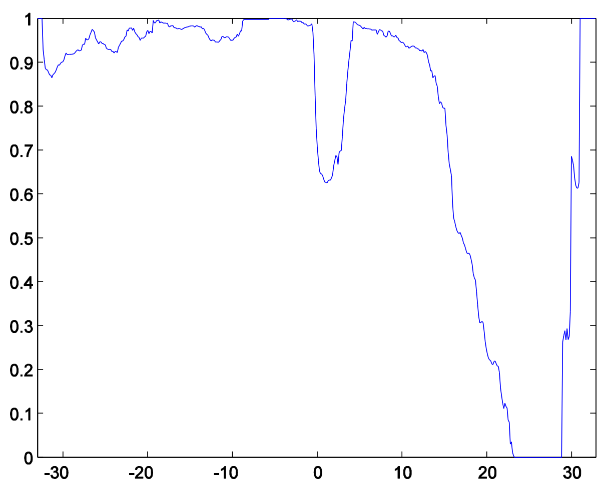

Figure 2: 1D Objective Function for Obstacle Avoidance from Stereo

This function can be regarded as a one-dimensional objective function for obstacle avoidance from stereo input, considering only the viewing / steering angle. This one dimensional objective function can then be extended for velocity as well, using the following formulation:

$$o_{stereo}(v,\alpha) = \frac{f(\alpha)}{1 + \left| v f(\alpha) / c \right|}$$

3. *Go To Goals*. The goal seeking behavior is assigned two tasks. First, it points the robot to the goal position and it varies the velocity respective to the distance to the goal. This means the development of the objective function can be split up as $o_{GoToGoal}(v,\alpha) = o^{\alpha}_{GoToGoal}(\alpha).o^{v}_{GoToGoal}(v)$. To calculate these objective functions, the (Euclidian) distance to the goal d_{Goal} and heading to this goal θ are calculated from the current robot position given by the GPS system and the current waypoint given by the global path planner. The goal seeking behavior aims to minimize the difference between the robot heading α and the goal heading θ, which can be formulated as:

$$o^{\alpha}_{GoToGoal} = \frac{1}{1+\left(\frac{\alpha-\theta}{\beta}\right)^2}$$

with β the window size which is considered. $o^{v}_{GoToGoal}(v)$ is set up such that the velocity is always high, with the exception that when the robot approaches a goal position, the speed should be reduced. This is expressed as:

$$o^{v}_{GoToGoal}(v) = \begin{cases} \left(\frac{v}{v_{\max}}\right)^2 & \text{if} \quad d_{Goal} > d_{Threshold} \\ \frac{1}{1+\left(\frac{v}{v_{\max}}\right)^2} & \text{if} \quad d_{Goal} < d_{Threshold} \end{cases}$$

4. Behavior Fusion

4.1. The multiple objective decision making problem

The 3 behaviors discussed above must be fused together to form one consistent and globally optimal robot command, to be sent to the robot actuators. The performance of the behavior-based controller depends on the implementation of the individual behaviors as well as on the method chosen to solve the behavior fusion or action selection problem. We have chosen a method to solve the action selection problem, by formulating it as a multiple objective decision making problem, as proposed by Pirjanian in [7]. Mathematically, a multiobjective decision problem can be represented as finding the solution to $\arg\max_{\mathbf{x}} \; o_1(\mathbf{x}),\ldots,o_n(\mathbf{x})$.

4.2. Classical methods for solving the multiple objective decision making problem

A common method for solving a multiple objective decision making problem is the weighting method. This method is based on scalar vector optimization and is formulated in the following way:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in X} \sum_{i=1}^{n} w_i o_i(\mathbf{x}) \quad ,$$

where w_i is a set of weights, normalized such that $\sum_{i=1}^{n} w_i = 1$.

An important question is how to vary the weights w_i to generate the whole or a representative subset of the solutions. Most approaches described in the literature handle the problem in the same manner. Each weight w_i is considered to have a reasonable number of discrete values between 0 and 1. Then the equation is solved for each combination of values. The obvious limitation of this approach is the large number of computations to be performed, as the computational complexity grows exponentially with the number of objective functions.

The interval criterion weights method [9] reduces the number of computations by incorporating some subjective knowledge in the process of generating optimal solutions. Basically, rather than letting each w_i assume values in the interval [0; 1], a narrower interval is used that reflects the importance of each objective function. Apart from reducing the computational burden, the interval criterion weights method, produces the `best' part of X*. Here 'best' reflects the decision maker's preferences which are expressed as weight values. A further advantage of the interval criterion weights method is due to the fact that in practice a decision maker seems to be more comfortable defining the range within which a weight value should be, rather than estimating the single weight values.

Goal programming methods [6] define a class of techniques where a human decision maker gives his/her preferences in terms of weights, priorities, goals, and ideals. The concept of best alternative is then defined in terms of how much the actual achievement of each objective deviates from the desired goals or ideals. Further, the concept of best compromise alternative is defined to have the minimum combined deviation from the desired goals or ideals. Goal programming methods thus choose an alternative having the minimum combined deviation from the goal $\hat{o}_1, \ldots, \hat{o}_n$, given the weights or priorities of the objective functions. According to the goal programming theorem, the solution to the multi-objective decision making problem can be expressed as:

$$\arg\min_{\mathbf{x} \in X} \sum_{i=1}^{n} w_i \left| o_i(\mathbf{x}) - o_i^* \right|^p$$

where $1 \leq p \leq \infty$, $\mathbf{o}^*$ is the ideal point, w_j is the weight or priority given to the j^{th} objective.

In [4], we proposed a method to choose the weights based upon a reliability measure associated to each behavior. The principle behind the calculation of the activity levels is that the output of a behavior should be stable over time in order to trust it. Therefore, the degree of relevance or activity is calculated by observing the history of the output of each behavior. This history-analysis is performed by comparing the current output $\varpi_{\varpi_{j,k}}^{b_i}$ to a running average of previous outputs, which leads to a standard deviation, which is then normalized. For a behavior b_i with outputs ϖ_j these standard deviations $\sigma_{\varpi_j}^{b_i}$ are:

$$\sigma_{\varpi_j}^{b_i} = c_{\varpi_j} \sum_{k=i-h}^{i} \left(\varpi_{\varpi_{j,k}}^{b_i} - \frac{\sum_{l=1}^{N} \varpi_{\varpi_{j,l}}^{b_i}}{N} \right)^2$$

The bigger this standard deviation, the more unstable the output values of the behavior are, so the less they can be trusted. The same approach is followed all behaviors. This leads to an estimate for the activity levels or weights for each behavior:

$$w_{b_i} = \sum_{j=1}^{number\ of\ outputs} 1 - \sigma_{\varpi_j}^{b_i}$$

For stability reasons, the activity level is initialized at a certain value (in general 0.5) and this estimate is then iteratively improved.

4.3. Proposed approach for solving the multiple objective decision making problem

The approaches towards solving the multiple objective decision making problem for action selection which we have reviewed up to this point all have their advantages and disadvantages. Solving the multiple objective decision making problem using reliability analysis has the big advantage of incorporating direct information from the system under control into the control process. On the other hand, this architecture does not offer a human decision maker the ability to interact with the decision process. As autonomous agents more and more have to interact with humans on the field, exchanging knowledge and learning from each other, this is a serious shortcoming. Common techniques for solving the multiple objective decision making problem while taking into account a decision maker's preferences take into account these issues by offering a human operator the ability to input some objectives or ideal points. These approaches, however, suffer from the disadvantage that no reliability data from the sensing and other processes is taken into account while performing action selection. One could thus argue that while reliability analysis-based approaches are too robot centric, these second set of approaches is too human-centric. Here, we propose an approach to integrate the advantages of both theorems. This can be achieved by minimizing the goal programming and reliability analysis constraints in an integrated way, following:

$$\arg\min_{\substack{\mathbf{x}\in X \\ w_i \in \mathbf{w}}} \left[\lambda \left(\sum_{i=1}^{n} w_i \left| o_i(\mathbf{x}) - o_i^* \right|^p \right) + 1 - \lambda \sum_{i=1}^{n} \left| w_i - \sum_{j=1}^{number\ of\ outputs} 1 - \sigma_{\varpi_j}^{b_i} \right|^p \right]$$

with λ a parameter describing the relative influence of both constraints. This parameter indirectly influences the control behavior of the robot. Large values of λ will lead to a human-centered control strategy, whereas lower values will lead to a robot-centered control strategy. The value of λ would therefore depend on the expertise or the availability of human experts interacting with the robot.

It is obvious that this method increases the numerical complexity of finding a solution, but this does not necessarily leads to increased processing time, as the search interval can be further reduced by incorporating constraints from both data sources.

5. Results & Conclusions

In this paper we have discussed three main aspects of the development of a behavior based control architecture for an autonomous crisis management robot in the framework of the View-Finder project. First, we presented a comprehensive and framework, which is extremely modular, thanks to the choice of the CoRoBa-based software architecture and the behavior-based control architecture. Secondly, we presented how different behaviors can be individually designed. Thirdly, we presented a novel technique of solving the behavior fusion or action selection problem by integrating the classical goal-programming method with a new approach based on reliability analysis.

As a result, the robot becomes a semi-autonomous agent: it can handle a high-level task (searching for human victims) with minimal input from human operators, by navigating in a complex, dynamic and environment, while avoiding potentially hazardous obstacles. If required, a remote human operator can still take control of the robot via the joystick, but in normal operation mode, the robot navigates autonomously to a list of waypoints, while avoiding obstacles (thanks to the stereo-based terrain traversability estimation) and while searching for human victims.

6. References

[1] E. Colon, H. Sahli and Y. Baudoin, "CoRoBa, a multi mobile robot control and simulation framework", International Journal of Advanced Robotic Systems, Vol. 3, Number 1, pp. 073-078, March 2006.

[2] G. De Cubber, D. Doroftei, L. Nalpantidis, G. C. Sirakoulis, A. Gasteratos. Stereo-based Terrain Traversability Analysis for Robot Navigation. Third International Workshop on Robotics for risky interventions and Environmental Surveillance-Maintenance, RISE'2009- Brussels, Belgium. January 2009.

[3] G. De Cubber, G. Marton. Human Victim Detection. Third International Workshop on Robotics for risky interventions and Environmental Surveillance-Maintenance, RISE'2009- Brussels, Belgium. January 2009.

[4] D. Doroftei, E. Colon, G. De Cubber, A behaviour-based control and software architecture for the visually guided Robudem outdoor mobile robot, , ISMCR2007, Warsaw, Poland, 21-23 June 2007

[5] J.L. Jones, "Robot Programming: A practical guide to Behavior-Based Robotics", 2004.

[6] H. Nakayama, Y. Yun: Generating Support Vector Machines Using Multi-Objective Optimization and Goal Programming. Multi-Objective Machine Learning 2006: 173-198

[7] P. Pirjanian. "Behavior coordination mechanisms - state-of-the-art". Tech. Rep. IRIS-99-375, Institute of Robotics and Intelligent Systems, School of Engineering, Univ. of South California, Oct. 1999.

[8] D. Slama, J. Garbis and P. Russell, "Enterprise CORBA", Prentice Hall, 1999

[9] Steuer, R. E. (1974). "Interval Criterion Weights Programming: A Portfolio Selection Example, Gradient Cone Modification, and Computational Experience." In Golden, M. and R. L. Schumacker (eds.), Proceedings: Tenth Southeastern Institute of Management Sciences Meeting, Clemson University Press, pp. 246-255.